Department of Computer Science	**Operating Systems**
VU Amsterdam	**20.12.2018**

This exam consists of one page. No calculator, pencil, or open books allowed. Concise answers!

1 What is the difference between system calls and C library calls? Can a full program be written by using only system calls and no C library calls? If so, what are the consequences? *10pt*

2 Using a sample execution run, show how the following concurrent program can come to a deadlock with N unconsumed items. *15pt*

```
1  process producer()
2  {
3    while(true)
4    {
5      produce_item();
6      if (count == N)
7        sleep();
8      enter_item();
9      count = count + 1;
10     if (count == 1)
11       wakeup(consumer);
12   }
13 }
```

```
1  process consumer()
2  {
3    while(true)
4    {
5      if (count == 0)
6        sleep();
7      remove_item();
8      count = count - 1;
9      if (count == N-1)
10       wakeup(producer);
11     consume_item();
12   }
13 }
```

3 What are the differences and similarities between signals and interrupts? Can an interrupt be raised in response to a signal? Can a signal be raised in response to an interrupt? Support your answer with examples. *15pt*

4 Least-recently used (LRU) page frames are often best to be evicted from memory, if there is a need to do so. Describe a technique that approximates keeping track of such page frames. Also give an example in which your technique may result in nonoptimal page replacement decisions. *15pt*

5 Intel is planning to support a 5-level page table organization in future x86 architectures, as an upgrade to the current 4-level page table organization on x86-64 (assume the page table format and the page table entry format stay the same). What benefits would this upgrade provide and for what classes of applications? What classes of applications, on the other hand, may experience degraded performance after the upgrade and why? What strategy would you recommend Intel to reduce the risk of such performance degradation? *15pt*

6 What disk operations are needed to fetch the i-node for a file with a `/usr/ast/courses/os/handout.t` path name (assume that the i-node for the root directory is in memory, nothing else along the path is in memory, all the directories fit in one disk block, all the inodes fit in one disk block, and the filesystem uses a typical i-node organization with both direct and indirect blocks)? How would your answer change if all directories fit in $N = 2$ disk blocks instead? Generalizing, how would your answer change as we increase N? Support your answer with an example. *15pt*

7 A slight modification of the elevator algorithm for scheduling disk requests is to always scan in the same direction. Can this modified algorithm improve the request processing latency (aka turnaround time) of the elevator algorithm? Why? Can the modified algorithm improve the throughput (aka number of requests per second) of the elevator algorithm? Why? *15pt*